Shaffer, T.L.; N.P. Groninger. Double data entry for novice users. In: Proceedings, 20th annual SAS Users Group International Conference; 2-5 April 1995; Orlando, FL. Cary, NC: SAS Institute, Inc. 1111-1116.

DOUBLE DATA ENTRY FOR NOVICE USERS

Terry L. Shaffer, National Biological Service, Northern Prairie Science Center, Jamestown, ND
N. Paige Groninger, Animal & Plant Health Inspection Service, Denver Wildlife Research Center, Denver, CO

ABSTRACT

We present a SAS/AF™ application for validating data entry using double-entry techniques. Our application is designed with the non-SAS™ programmer in mind and does not require manipulating SAS code to tailor the application to a particular data set. You define the variables, enter observations into an initial data set, and then reenter to validate. During validation, each value entered is compared against the corresponding value from the initial data set. If discrepancies occur, the field is flagged and you are prompted to enter the correct value. The application keeps track of validated observations so that the validation process can span multiple data entry sessions. The application also allows for the possibility that observations may be omitted or duplicated.

INTRODUCTION

Double-entry of data is recognized as an effective method for reducing data entry errors. Boyle and Brinsfield (1993) provided a double data entry application using SAS/FSP™ software. The user enters the data once to create what Boyle and Brinsfield refer to as a master data set and then reenters the data into a separate data set. As entries are made to the second data set, Screen Control Language (SCL) is used to compare them with corresponding entries in the master data set. When a discrepancy occurs, the user is prompted to enter the correct value before leaving the field. If the error is in the master data set, the master data set is automatically updated by the SCL. McDonald (1992) developed an easy-to-use double-entry routine, but the two data sets were not compared until all observations had been entered twice. Both the Boyle and Brinsfield and the McDonald applications have limitations.

Boyle and Brinsfield made several assumptions about the data entry process. First, data must be entered in the same sequence in both data sets. This poses no serious limitations and is probably necessary for any double-entry process in which validation is immediate. They also assumed that the master data set contained no duplicate observations and that no observations were omitted. These assumptions are unrealistic for data sets with many observations. In addition, the Boyle and Brinsfield application is inflexible. Modifying it to work with data sets having different variables requires the user to edit and recompile SCL program statements.

The McDonald application did not require the user to make changes to the program. Unlike the Boyle and Brinsfield application, however, validation was not immediate. McDonald's approach, although effective, is inefficient; when a discrepancy is encountered the user must often search through many data sheets to find the correct value. In addition, we have observed that McDonald's application does not always detect duplicate or missing observations in one or both data sets.

FEATURES OF AN IMPROVED DOUBLE-ENTRY APPLICATION

In research environments like ours, the need exists for a generic double-entry application for use by scientists and technicians with limited knowledge of SAS and no knowledge of SCL. The application must allow the user to specify the data set structure, including variable names and attributes, and validate data without making changes to the program code. In addition, validation must occur immediately after the observation is entered the second time. To allow for the possibility of duplicate observations in the first data set, the application must permit those observations to be deleted during validation. Similarly, the application must allow observations to be added during validation to accommodate omitted observations. Because many data sets are too large to be validated during one data entry session, the application must allow the user to resume validation at the appropriate observation.

ANATOMY OF THE IMPROVED APPLICATION

The flowchart in Fig. 1 depicts the process for entering and validating data with our application. Two data sets are involved; we refer to these as the "initial" and "validation" data sets. After validation is completed, the initial data set should be deleted. The correct values are stored in the validation data set.

GETTING STARTED

To begin the process, our application prompts you to enter a valid SAS data set name, which must be seven characters or less. This names the validation data set. Another data set name, formed by adding an "X" to the end of the previous name, is used for the initial data set. If the initial data set does not exist, a message is displayed informing you that the data set will be created. The SCL function CALL NEW then

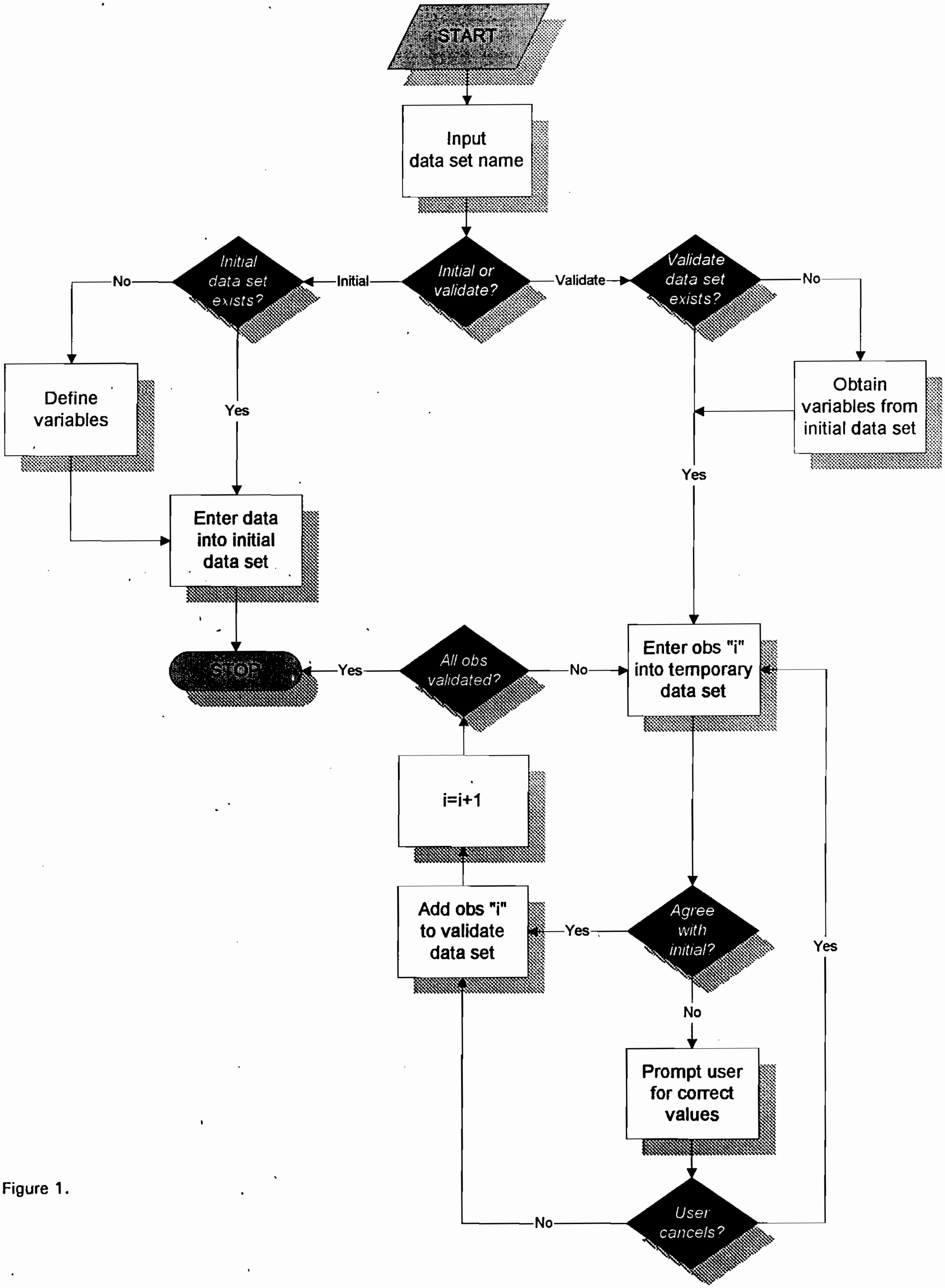
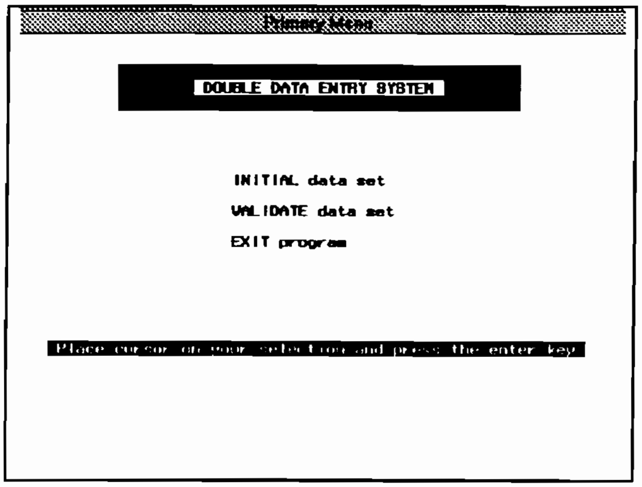

START
Input data set name
Initial or validate?
Initial
Validate
Initial data set exists?
No
Yes
Define variables
Enter data into initial data set
Validate data set exists?
No
Obtain variables from initial data set
Yes
Enter obs "i" into temporary data set
STOP
Yes
All obs validated?
No
i=i+1
Add obs "i" to validate data set
Yes
Agree with initial?
No
Prompt user for correct values
User cancels?
No
Yes

Figure 1.

displays a data definition screen allowing you to define variables and their attributes. If the initial data set exists, a menu is invoked (Fig. 2) allowing you to choose between editing the initial data set and validating observations.



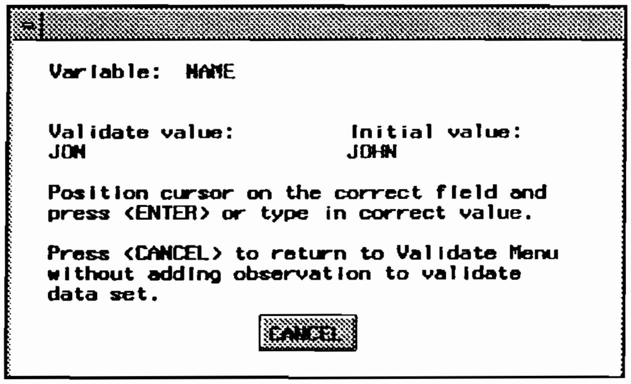
Place cursor on your selection and press the enter key

Figure 2.

VALIDATING DATA USING DOUBLE-ENTRY

Like Boyle and Brinsfield, we use the FSEDIT procedure of SAS/FSP software for entering both initial and validation data sets. However, we do not use SCL with our FSEDIT screens to compare values from initial and validation data sets, because our screens must work with generic data for which variable names and attributes are unknown at SCL compile time.

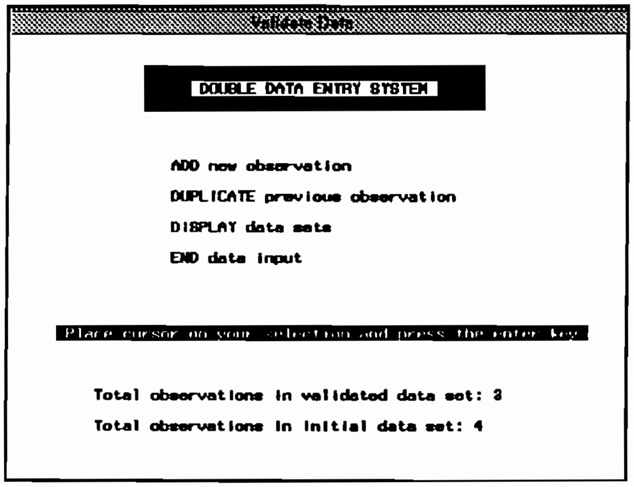
To validate an observation, you reenter it into a 1-observation temporary data set using FSEDIT. Once the observation is reentered, you issue the END command to pass control to the SAS/AF program entry "validate.program." In this program, we use SCL to determine how many variables are in the data set and each variable's type. Depending on whether the type is character or numeric, we pass control to either "errorc.program" or "errorn.program." These programs compare the variable's values from the initial and temporary data sets. If the values match, we add the observation to the validation data set. If the values do not match, we display the variable name and values from each data set (Fig. 3). You then choose one of the two values or enter a new value. If you are unsure of the correct value, you may issue CANCEL to abort validating that observation. Once you have validated all variables, the observation is added to the validation data set.



```
Variable:  NAME

Validate value:          Initial value:
JON                      JOHN

Position cursor on the correct field and
press <ENTER> or type in correct value.

Press <CANCEL> to return to Validate Menu
without adding observation to validate
data set.
```

Figure 3.

After the validation data set is updated, a menu is displayed allowing you to continue validating by either adding a new observation to the temporary data set or by duplicating the previous observation (Fig. 4). The menu includes a status line indicating the number of observations in the initial data set and the number that have been validated. Validation continues until you decide to quit or you have validated all observations in the initial data set.



Place cursor on your selection and press the enter key

Total observations in validated data set: 3

Total observations in initial data set: 4

Figure 4.

DEALING WITH DUPLICATE OBSERVATIONS

Duplicate observations are a common type of data entry error. Duplicates usually occur consecutively, and cause loss of data set synchronization during validation. When discrepancies begin to occur on successive observations, the reason may be a duplicate observation in the initial data set (see Table 1). To help in identifying duplicates, our application allows you to view the initial and validation data sets in adjacent displays using the FSVIEW procedure (Fig. 5). You invoke these displays by choosing "DISPLAY data sets" from the "Validate Data" menu (Fig. 4). You can delete a duplicate in the initial data set by placing a "d" in the action variable (ACTN) for that observation.

Table 1. Data sets in which observation 2 has been duplicated in the initial data set.

Initial Data Set		Validation Data Set	
OBS	NAME	OBS	NAME
1	Barbara	1	Barbara
2	John	2	John
3	***John***	3	Mike
4	Mike	4	Tom
5	Tom		

DEALING WITH MISSED OBSERVATIONS

Just as observations may be duplicated in the initial data set, they may also be omitted (see Table 2). Missed observations also cause the validation process to go awry. This application will not allow more observations in the validation data set than exist in the initial data set. As in dealing with duplicates in the initial data set, choose "DISPLAY data sets" from the "Validate Data" menu to view the initial and validation data sets side-by-side. If an observation is missing from the initial data set, place an "i" in the ACTN variable on the observation immediately before the missed observation (Fig. 6). This displays a blank observation so you may enter values for the missed observation. Validation will resume at the point of the inserted observation; any previously validated observations following that observation are no longer considered validated and must be reentered. If order of the observations in the validation data set is not important, we recommend that insertions be made at or beyond the last validated observation to avoid having to revalidate.

Table 2. Data sets in which observation 3 has been omitted from the initial data set.

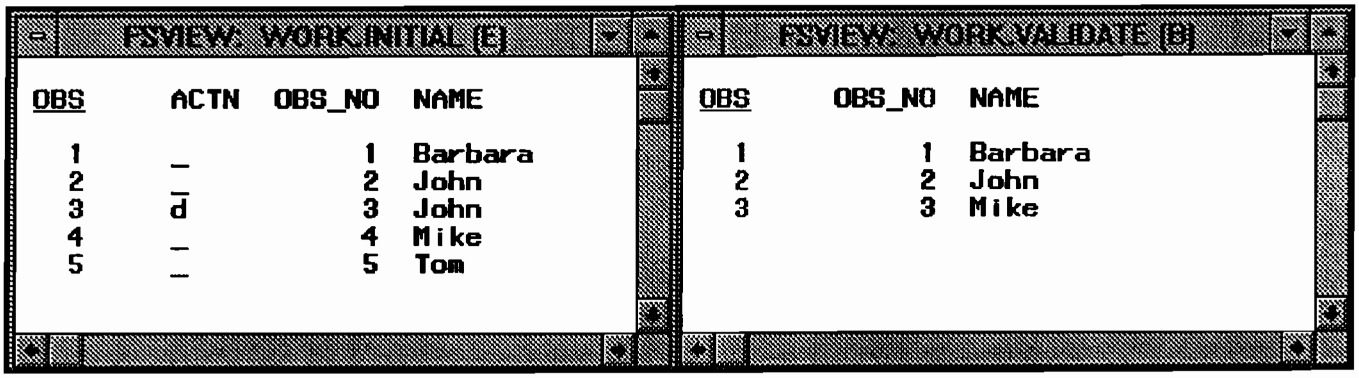
Initial Data Set		Validation Data Set	
OBS	NAME	OBS	NAME
1	Barbara	1	Barbara
2	John	2	John
3	Tom	***3***	***Mike***
		4	Tom

PICKING UP WHERE YOU LEFT OFF

Because the double-entry application keeps track of how many observations have been validated, the validation process can span multiple data entry sessions. With large data sets, you may want to enter a subset of the data into the initial data set, validate that subset, then repeat the process. To improve performance when entering large data sets, you may want to enter these subsets into separate data sets and combine them following validation. Procedures for deleting and inserting observations, and for displaying the initial and validation data sets side-by-side, require processing every observation in both data sets. This processing, although minimal, may result in unsatisfactory performance with large data sets on slower systems.

HELP SCREENS, PMENUS AND FUNCTION KEYS

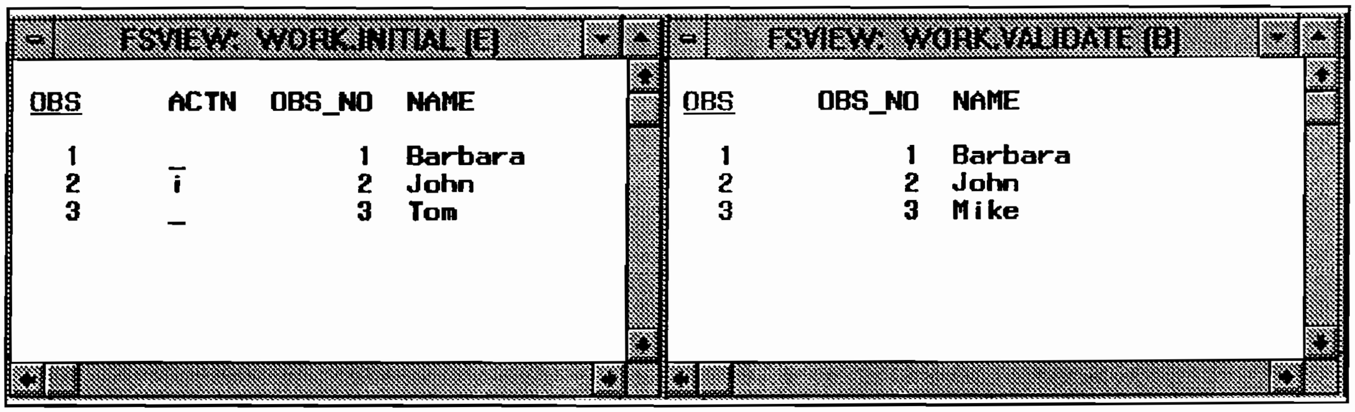
All SAS/AF and SAS/FSP entries are assigned custom HELP screens and pop-up menus. Pop-up menus are activated by pressing the right mouse button or F1 function key. All other function keys have been disabled to avoid inadvertently issuing commands that would interfere with or confuse the application. Each pop-up menu item is assigned a mnemonic, generally the first letter of the command.



OBS	ACTN	OBS_NO	NAME
1	_	1	Barbara
2	_	2	John
3	d	3	John
4	_	4	Mike
5	_	5	Tom

OBS	OBS_NO	NAME
1	1	Barbara
2	2	John
3	3	Mike

Figure 5.



OBS	ACTN	OBS_NO	NAME
1	_	1	Barbara
2	i	2	John
3	_	3	Tom

OBS	OBS_NO	NAME
1	1	Barbara
2	2	John
3	3	Mike

Figure 6.

CUSTOMIZING FSEDIT

With FSEDIT, you can customize the data entry screen by relocating and relabeling variables. In addition, you can make other changes, including specifying initial values, providing a range of allowable values, controlling justification, changing field color, and controlling case-sensitivity of character variables. More knowledgable users can further customize the data entry process by writing SCL code specific to their data set. In our application, FSEDIT stores the custom settings in a permanent screen catalog with the same name as the validation data set, except for the file name extension. Like all Release 6.08 and later catalogs, the screen catalog file has the extension .sc2, whereas the validation data set has a .sd2 extension. You can access the screen customization features of FSEDIT by choosing the "Customize" option from within FSEDIT when entering the initial data set. Help with screen customization can be found in the SAS/FSP manual (SAS Institute Inc. 1989).

ASSUMPTIONS

We assume that data are entered in the same sequence for both data sets. Although we include numerous error-trapping routines and have tested the application thoroughly, to quote McDonald (1992), we implicitly assume "... cooperative, non-malicious, and intelligent use."

The application was developed and tested using Release 6.08 of The SAS System for Microsoft Windows™ but should work with any operating system running Release 6.08 or later. We have used it successfully with both releases 6.08 and 6.10 of the SAS System for OS/2™.

CONCLUSIONS

Our goal was to develop a generic double-entry application that scientists and technicians with limited knowledge of SAS and no knowledge of SCL could use. Our application is generic; it works with data sets having any number of variables. Variables can have any name, length or type as long as they are valid for The SAS System. Users do not need to modify SAS code to use the application with their data. They will, however, need a basic understanding of how data are represented as observations and variables.

Depending on their requirements, users may need to customize FSEDIT. For example, FSEDIT automatically converts all character variables to upper-case. It can be configured to be case-sensitive, but this requires user customization. Although not particularly difficult, screen customization could be confusing for inexperienced users. Knowledge of SCL is not necessary, but could be an asset in certain situations.

Our application allows for the possibility that observations may be omitted or duplicated. By deleting and adding observations in the initial data set, the user can resynchronize the initial and validation data sets. Also, initial data entry and validation can span multiple sessions. In summary, the application provides researchers and others a flexible, easy-to-use tool for entering and validating data.

ACKNOWLEDGMENTS

We thank R. E. Engeman, H. W. Krupa, and M. D. Schwartz for comments and suggestions on earlier drafts of this paper.

REFERENCES

Boyle, J. and C. Brinsfield. 1993. "Beyond journaling: data validation with dual data entry," *Observations*™: the technical journal for SAS software users. 2(3), pp. 4-12. SAS Institute Inc., Cary NC.

McDonald, T. 1992. "An easy-to-use double-entry routine," Proceedings of the seventeenth annual SAS users group international conference. pp. 893-898. SAS Institute Inc., Cary NC.